06-6-05

AFF
IC

PATENT
450117-4840

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant(s)	:	Yoeri APTS et al.
Serial No.	:	09/246,271
For	:	**METHOD AND SYSTEM FOR COMMUNICATION BETWEEN APPLICATION PROGRAMS AND A NETWORK**
Filed	:	February 8, 1999
Examiner	:	HO, Andy
Art Unit	:	2126

745 Fifth Avenue
New York, NY 10151

EXPRESS MAIL

Mailing Label Number: ED 521638049 US

Date of Deposit: June 3, 2005

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Chiaki Kokka

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

APPEAL BRIEF OF APPELLANT

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal from the Final Rejection by the Examiner dated February 9, 2005, which issued on the above-identified application of claims 1, 2 and 4-22. This Brief is submitted in triplicate. The requisite fee set forth in 37 C.F.R. §1.17(c) has been paid.

06/07/2005 MAHMED1 00000054 09246271
01 FC:1402 500.00 OP

00283307

REAL PARTY IN INTEREST

The real party in interest in this appeal is Sony Europa B.V., a Netherlands corporation, with offices at Schipholweg 275, Badhoevedorp, Netherlands NL-1171, to which appellant has assigned all interest in, to and under this application, by virtue of an assignment recorded on April 3, 2003 at Reel 13536, Frame 0494 of the assignment records of the Patent and Trademark Office.

RELATED APPEALS AND INTERFERENCES

Upon information and belief, the undersigned attorney does not believe that there is any appeal or interference which will directly affect, be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF THE CLAIMS

Claims 1, 2, 5-8, 12, 15-16, 18, and 21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwaderer (U.S. Patent No. 6,393,496; hereinafter referred to as "Schwaderer") in view of Kanamori (U.S. Patent No. 6,338,079). Claims 9-11, 17, 20 and 22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwaderer in view of Kanamori, and further in view of Tanenbaum (Network Architecture, 1992 publication). Claims 4, 13, and 19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwaderer in view of Kanamori, and further in view of Jardine (U.S. Patent No. 5,619,647). Claim 14 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwaderer in view of Kanamori, and further in view of Phillips (U.S. Patent No. 6,289,393).

The status of the claims are as follows:

Claims allowed: none

Claims objected to: none

Claims rejected: 1, 2, and 4-22

The rejected claims 1, 2, and 4-22 are set out in the Appendix.

Appellant is appealing the Final Rejection of claims 1, 2, and 4-22, which constitute all of the currently pending claims in this application.

STATUS OF THE AMENDMENTS

The last amendments filed in this case were an Amendment and Supplemental Amendment filed by the Appellant on September 3, 2004, prior to the issuance of the Final Office Action in this case on February 9, 2005. No Amendment after Final was filed.

SUMMARY OF THE INVENTION

The present invention is directed to a method and system for communication between an application program and a network device driver program through intermediate structure software in an Object-Oriented Operating System (OS) that allows for object-oriented programming.

As explained in the Specification, within this system, data units travel through a set of interconnecting modules. (Specification, p. 12, lines 7-8.) The basic data unit that is used to pass information between the modules is called the Service Data Unit (SDU). (Specification, p. 9, lines 20-22.) SDUs are dynamic memory buffers, shared by all modules, used for data manipulations whereby physical copying of data is avoided as much as possible. (Specification, p. 9, lines 22-24.) Since all these manipulations need to be performed as fast as possible, data units are not necessarily copied during this process. (Specification, p. 12, lines 10-12.) Instead data references pointing to data units and queues are passed between the modules. (Specification, p. 12, lines 13-14.)

Therefore all SDUs are managed by a central SDU Manager which manages a memory pool of data units. (Specification, p. 12, lines 14-15.) This memory pool is shared between the SDU manager and all related modules and queues. (Specification, p. 12, lines 14-15.) Figure 4 below demonstrates how SDUs 23 and 24 of program objects 20 and 21 are in the memory pool 22, which is managed by the SDU Manager 25.

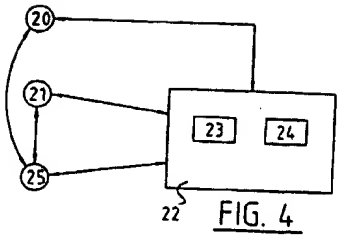

FIG. 4

To avoid physical copying of the SDUs when moving from one program object to another, the SDUs are stored using references, which references point to the memory location of the SDU, and offsets and sizes of the data units within the SDU, as is shown in figures 5-7. (Specification, p. 12, lines 22-25.) Therefore, the SDUs are organized in SDU pools, which are shared memory buffers. (Specification, p. 13, lines 25-27.)

The memory formats and logical formats of the data units are shown in Figs. 5a-7b of the subject application, which are reproduced below:

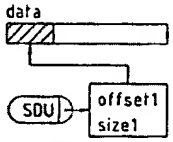

FIG. 5a

FIG. 5b

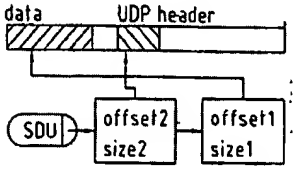

FIG. 6a

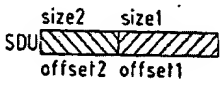

FIG. 6b

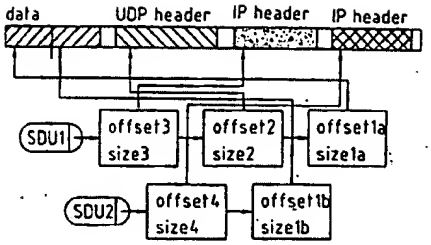

FIG. 7a

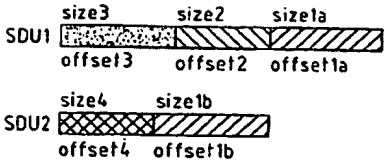

FIG. 7b

These figures demonstrate, for example, how a program application sends data units over an ATM network using the User Datagram Protocol (UDP). (Specification, p. 12, lines 26-28.) The application program first builds an SDU that contains the application data units. (Specification, p. 12, lines 28-29.) The original SDU memory layout of memory pool 22 is shown in figure 5a and the logical structure thereof is shown in figure 5b. (Specification, p. 12, lines 30-32.) As shown in these figures, the data contained in the SDU is stored in variable sized data units and the actual SDU is constructed by linking data together at certain offsets. (Specification, p. 12, lines 32-35.)

The SDU is then sent to the UDP module, which adds the appropriate UDP header in front of the SDU. (Specification, p. 12. lines 35-37.) Fig. 6a illustrates the resulting SDU memory-layout and figure 6b illustrates the resulting logical structure. (Specification, p. 12, lines 37-38 & p. 13, lines 1-2.)

The UDP module then passes the SDU to the IP module, which fragments the UDP datagram if needed and adds an IP header to every fragment. (Specification, p. 13, lines 2-5.) The effects of fragmentation and IP header insertion on the SDU is shown in figures 7a and 7b. (Specification, p. 13, lines 5-7.)

Data units are then passed to the ATM driver that sends it over the network. Throughout these manipulations the SDU data unit is never copied or moved. (Specification, p. 13, lines 7-9. Emphasis added.) This makes the communication method fast and reliable. (Specification, p. 13, lines 12-13.)

Managing the SDUs as described above has the following advantages:

(1) data can be shared between SDUs, that is copying or moving of data from one SDU to another is done by keeping references and reference counters instead of physically copying or moving the data;

(2) adding data and removing data from an SDU can be accomplished without physically copying or moving the SDU contents;

(3) SDU data does not have to be stored contiguously in memory. (Specification, p. 13, lines 14-24.)

Therefore, some important aspects of the present invention are: 1) passing references to data rather than the data itself, to achieve a fast and reliable data transfer time; 2) storing data, to be pointed to by the references, in SDU pools accessible by the different program objects or modules.

THE ISSUES PRESENTED

The following issues are presented in this appeal:

1. Whether claims 1, 2, 5-8, 12, 15-16, 18, and 21 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori?
2. Whether claims 9-11, 17, 20, and 22 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori, and further in view of Tanenbaum?

3. Whether claims 4, 13, and 19 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori, and further in view of Jardine?

4. Whether claim 14 is unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori, and further in view of Phillips?

GROUPING OF THE CLAIMS

It is the Appellant's intention that claims 1, 2 , and 4 - 22 stand or fall together.

ARGUMENTS

1. Whether claims 1, 2 , 5-8, 12, 15-16, 18 and 21 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori?

Claims 1, 2, 5-8, 12, 15-16, 18, and 21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwaderer (U.S. Patent No. 6,393,496; hereinafter referred to as "Schwaderer") in view of Kanamori (U.S. Patent No. 6,338,079).

Independent claim 1 recites the following:

> 1. (Currently Amended) Method for communication between an application program and a network device driver program through intermediate structure software, comprising the steps of:
> a. supplying of application data units from the application program to a first program object being part of the intermediate structure software;
> b. performing of first functions of the first program object on the application data units;
> c. supplying of resulting first data units from the first program object to a second program object being part of the intermediate structure software;
> d. performing of second functions of the second program object on the first data units;
> e. supplying of the resulting second data units to the network device driver program;
> wherein supplying data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and
> wherein for at least one application data unit, the referenced memory location storing data of the application data unit is the same memory location as the referenced memory location storing at least some of the data of the corresponding first data unit and as the referenced memory location for storing at least some of the data of the corresponding second data unit.

From the claim language above, it is clear that the memory location storing data of the application data unit is pointed to by a reference passed between the program objects.

This same memory location, pointed to by a reference passed between the program objects, also stores data of the corresponding first data unit and at least some data corresponding to the second data unit. In sum, one memory location stores data from each of three sets of data units, and this same memory location is pointed to by references passed between program objects.

In his rejection of independent claim 1 under section 103 in the Final Office Action, the Examiner states that Schwaderer discloses all of Claim 1's elements except for the claimed technique of passing references and memory locations of the data units. (Final Office Action, page 3.) The Examiner argues that Kanamori teaches the claimed reference passing system; i.e., "a system (Column 1, line 20 to Column 2, line 14; Fig. 1) of passing data (col. 1, line 21) between a transferer (Col. 1, line 27) and a transferee (Col. 1, line 28) using a technique of passing references pointing to the memory location storing the data . . . (col. 1, lines 37-41) " (Final Office Action, p.3.) However, the Examiner does not appear to explain how this passage of Kanamori addresses keeping the data from each of three data units in the same memory location as called for in claim 1.

The only reference made by the Examiner to the "same memory location" element of claim 1 appears to be in the Examiner's discussion of Schwaderer on page 3 of the Office Action, wherein the Examiner states: "the memory locations of the data units are the same (lines 19-30 column 4)." Column 4, lines 19-30 of Schwaderer merely states, however:

> "The present invention also includes a media control layer functionally positioned between the operating system and the network. The media control layer receives operating system I/O calls from the operating system and translates the data or requests into a message or packet the network will understand. Use of the media control layer allows application program and operating system operation in a manner which is independent of the network requirements. A network device driver can also be included functionally positioned between network and the media control

> layer. The network device drive is responsible for the hardware communication control between the set top box and the network."

It is not clear how this passage of Schwaderer shows that the data for the data units is kept in the same memory location. This passage does not mention "memory" or "memory location," or "data units." Nor does it otherwise appear to address memory locations, much less the memory locations of data for the data units as called for in claim 1. Therefore, it is respectfully submitted that neither Schwaderer nor Kanamori, as applied by the Examiner, disclose storing data for the data units in the same referenced memory location, as called for in claim 1.

It is axiomatic that for a rejection under 35 U.S.C. § 103 there must be some teaching or suggestion in the prior art to make the claimed combination. The mere fact that references can be combined or modified does not render the resultant combination obvious over the prior art unless the prior art also suggests the desirability of the combination. In re Mills, 916 F.2d 680, 682 (Fed. Cir. 1990). If the proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. In re Gordon, 733 F.2d 900, 902 (Fed. Cir. 1984).

The Examiner argues that Kanamori teaches a "technique of passing references pointing to the memory location storing the data . . ." (col. 1, lines 37-41). (Final Office Action, page 3). Yet even if one were to interpret this portion of Kanamori as requiring data to be stored in the same memory location, this language does not, when combined with Schwaderer, render obvious the method of claim 1. In fact, Kanamori teaches away from providing a shared memory location for the data of the data units, as claimed. Kanamori describes that the intention of the invention as directly opposite a shared memory location. As described in the Summary of

the Invention section of Kanamori: "It is an object of the invention to provide a method and system . . . for providing a group of parallel resources as a proxy, or substitute, for a single shared resource." Abstract, col. 2, ll. 65-67. In other words, the aim and purpose of Kanamori is to provide a system in which the accessed memory locations are distinct and separate — not the same — as described in claim 1.

This motivation for different accessible memory locations is based on the prior problems with multiple programs accessing data in shared memory protocol systems. (Kanamori, Col. 2, ll. 25-49.) For example, in a non-global operating system, in which the system is unable to allocate a global resource (such as a memory block) that is accessible to every program, the system creates a putatively global resource. (Kanamori, Col. 3, ll. 28 – 35.) When the facility detects that a transferor program is attempting to share with a transferee program this putatively global resource, the facility creates a proxy, or substitute resource accessible to the transferee. (Kanamori, Col. 3, lines 36-40.) In other words, the transferee accesses a proxy memory block that is separate and distinct from memory accessible to the transferor program. If one were to modify Kanamori to accomplish the claimed aim of providing a shared referenced memory location, as claimed, one would render Kanamori unsatisfactory for its intended purpose. Thus, there is no motivation to combine Kanamori and Schwaderer to render obvious the system described in claim 1.

Further, the combination of Schwaderer and Kanamori teaches against the efficient data transfer process of claim 1. In Kanamori, actual data is copied prior to its access by the different program objects. References to data are not passed, as specified by claim 1. For instance, when creating a "proxy memory block," the facility "preferably copies the contents of the putatively global memory block to the proxy memory blocks." (Kanamori, Col. 3, ll. 46-49.)

In other words, in Kanamori an entire set of data within the memory block must be copied before it is accessed by a second program object. This process does not teach the present invention's advantage of decreasing the data transfer time by only passing references, in place of copying data. In sum, combining Kanamori and Schwaderer would teach away from the required "passing references" and "same memory location" elements of claim 1. As a result, the combination of Kanamori and Schwaderer, as referenced by the Examiner, would not render obvious the method of claim 1.

Accordingly, it is respectfully submitted that independent claim 1 is not anticipated by nor rendered obvious by the teachings of Schwaderer and Kanamori, as presented and referenced by the Examiner. Claims 2, 5-8, 12, and 21 depend from claim 1. Therefore, it is also submitted that the Examiner has not established how the cited combination of Schwaderer and Kanamori shows or suggests claims 2, 5-8, 12 and 21, through their dependence on claim 1. Similar arguments apply to claims 15 and 18, and so to claim 16 that depends from claim 15.

Based upon the foregoing, it is submitted that claims 1-2, 5-8, 12, 15-16, 18 and 21 are not anticipated by nor rendered obvious by the teachings of Schwaderer and Kanamori, as presented and referenced by the Examiner.

2. Whether claims 9-11, 17, 20, and 22 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori, and further in view of Tanenbaum?

Claims 9-11, 17, 20, and 22 depend from one of the independent claims 1 and 15. As discussed above, it is submitted that the cited combination of Schwaderer and Kanamori do not render claim 1 nor 15 obvious. Therefore, it is respectfully submitted that the rejection of

claims 9-11, 17, 20 and 22 has also been overcome through the dependence of these claims on one of the independent claims 1 and 15.

Furthermore, claim 20 calls for:

> 20. (Previously Presented) Method according to claim 10, wherein at least two data units referenced by a service data unit are stored in non-contiguous portions of memory.

On page 5 of the February 2, 2005 Office action, the Examiner indicates that Tanenbaum (last paragraph page 21 to line 3 page 22) teaches the data units are stored in non-contiguous portions of memory, as described in claim 20. Yet the section of Tanenbaum cited by the Examiner describes how an IDU (information data unit) is passed between entities across a network. It does not appear to address how data is stored or arranged in memory. Accordingly, it does not appear to address storing the data units referenced by a service data unit in non-contiguous portions of memory. For this reason as well, it is submitted that the cited portion of Tanenbaum does not appear to disclose or teach storing data units in "non-contiguous portions of memory" as claimed. Therefore, it is respectfully submitted that claim 20 is distinguishable from the applied or proper combination of Schwaderer, Kanamori, and Tanenbaum, as referenced by the Examiner.

In addition, claim 22 calls for:

> 22. (Previously Presented) Method according to claim 1, further comprising creating a service data unit for each application data unit, each service data unit including a size value indicating the size of data of the application data unit and an offset value indicating the memory location storing data of the application data unit,
>
> wherein supplying data units between program objects by passing references includes passing service data units corresponding to the supplied data units.

Accordingly, in one aspect of claim 22, the service data unit for an application data unit includes a size value and an offset value. The size value and offset value indicate the size and memory location, respectively, of the data of the application data unit, referenced by the service data unit. (See, e.g., Figure 5-7.) On page 6 of the Office Action, the Examiner argues: "Tanenbaum further teaches creating a service data unit (SDU, line 4 last paragraph page 21) with a size value and an offset value for each application data unit (second paragraph page 22)." However, the portion of Tanenbaum cited by the Examiner does not discuss a size value or offset value relating to the data of an application data unit at all. Rather, this section discusses how an SDU, as part of IDU, is sent across a network, along with control information. The control information may contain the number of bites of the SDU (service data unit), but the claimed size value must relate to "the size of data of the application data unit," not the service data unit. (See Claim 22.) Therefore, it appears the cited passage of Tanenbaum does not appear to teach or render obvious the "size value" limitation of claim 1.

Moreover, the cited passage of Tanenbaum does not mention "offset value" nor appear to discuss offset values at all. Therefore, it appears that Tanenbaum, as cited and argued by the Examiner, does not anticipate or render obvious an offset value "indicating the memory location storing data of the application data unit." (See claim 22.) Therefore, it appears that the SDU components described in claim 22 are not disclosed or suggested by the cited passage of Tanenbaum. Thus, the cited combination of Tanenbaum, Schwaderer and Kanamori as a whole do not suggest or render obvious claim 22.

Accordingly, it is respectfully submitted that that claims 9-11, 17, 20, and 22 are not anticipated by nor rendered obvious by the teachings of Schwaderer, Kanamori, and Tanenbaum, as presented and referenced by the Examiner.

3. Whether claims 4, 13, and 19 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori, and further in view of Jardine?

Claims 4, 13, and 19 depend from claim 1. As discussed above, it is submitted that the cited combination of Schwaderer and Kanamori do not render claim 1 obvious. Therefore, it is respectfully submitted that the rejection of claims 4, 13, and 19 has also been overcome through the dependence of claims 4, 13, and 19 on claim 1.

Based upon the foregoing, it is submitted that claims 4, 13, and 19 are not anticipated by nor rendered obvious by the teachings of Schwaderer, Kanamori, and Jardine, as presented and referenced by the Examiner.

4. Whether claim 14 is unpatentable under 35 U.S.C. § 103(a) as being obvious over Schwaderer in view of Kanamori, and further in view of Phillips?

Claim 14 depends from claim 1. As discussed above, it is submitted that the cited combination of Schwaderer and Kanamori do not render claim 1 obvious. Therefore, it is respectfully submitted that the rejection of claim 14 has also been overcome through the dependence of claim 14 on claim 1.

Based upon the foregoing, it is submitted that claim 14 is not rendered obvious by the teachings of Schwaderer, Kanamori, and Phillips, as presented and referenced by the Examiner.

CONCLUSION

Claims 1, 2, 5-8, 12, 15-16, 18, and 21 are not obvious in view of the applied or proper combination of Schwaderer and Kanamori. Accordingly, it is respectfully submitted that

the Examiner erred in rejecting claims 1, 2, 5-8, 12, 15-16, 18, and 21 and reversal of such rejections by this Honorable Board is solicited.

Claims 9-11, 17, 20, and 22 are not obvious in view of the applied or proper combination of Schwaderer, Kanamori, and Tanenbaum. Accordingly, it is respectfully submitted that the Examiner erred in rejecting claims 9-11, 17, 20, and 22 and reversal of such rejections by this Honorable Board is solicited.

Claims 4, 13, and 19 are not obvious in view of the applied or proper combination of Schwaderer, Kanamori, and Jardine. Accordingly, it is respectfully submitted that the Examiner erred in rejecting claims 4, 13, and 19 and reversal of such rejections by this Honorable Board is solicited.

Claim 14 is not obvious in view of the applied or proper combination of Schwaderer, Kanamori, and Phillips. Accordingly, it is respectfully submitted that the Examiner erred in rejecting claims 14 and reversal of such rejections by this Honorable Board is solicited.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP

By: Samuel S. Lee, Reg. No. 42,791 for
William S. Frommer
Reg. No. 25,506
(212) 588-0800

APPENDIX

CLAIMS ON APPEAL

1. Method for communication between an application program and a network device driver program through intermediate structure software, comprising the steps of:

a. supplying of application data units from the application program to a first program object being part of the intermediate structure software;

b. performing of first functions of the first program object on the application data units;

c. supplying of resulting first data units from the first program object to a second program object being part of the intermediate structure software;

d. performing of second functions of the second program object on the first data units;

e. supplying of the resulting second data units to the network device driver program;

wherein supplying data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and

wherein for at least one application data unit, the referenced memory location storing data of the application data unit is the same memory location as the referenced memory location storing at least some of the data of the corresponding first data unit and as the referenced memory location for storing at least some of the data of the corresponding second data unit.

2. Method according to claim 1, wherein data units are supplied over interconnecting queue-objects.

3. (Canceled)

4. Method according to claim 2, wherein data units are supplied over interconnecting queue-objects, wherein the queue-objects have different priorities.

5. Method according to claim 1, wherein program objects are added during run time of the application program.

6. Method according to claim 1, wherein program objects are removed during run time of the application program.

7. Method according to claim 1, wherein after performing of functions of a program object and supplying the data units to a further program object additional functions of the program object are performed.

8. Method according to claim 2, wherein step a and/or c also comprises adding or removing information to or from said data units.

9. Method according to claim 1, also comprising dividing data units into data units parts or uniting data unit parts into data units.

10. Method according to claim 1, providing service data units containing one or more data units.

11. Method according to claim 10, referencing data units with a reference to the service data unit.

12. Method according to claim 1, also providing a specialized execution environment for communication between the application program and the network device driver program.

13. Method according to claim 1, wherein data units are organized in data unit pools adapted to the specific use thereof.

14. Method according to claim 1, providing a naming service for mapping between the internal communication mechanism of the hardware and symbolic names.

15. System for communication between an application program and a network device driver program and vice versa through intermediate structure software, comprising.

a. a first program object being part of the intermediate structure software and for performing of first functions on data units, said data units being transferred to and from the application program and data units being transferred to and from said first program object;

b. a second program object being part of the intermediate structure software and for performing of second functions on said data units, said data units being transferred to and from said second program object and data units being transferred to and from the network driver;

wherein transferring data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and

wherein for at least one data unit, data of the data unit is not moved from the referenced memory location of that data unit to a different memory location while the first program object performs said first functions and while the second program object performs said second functions.

16. System according to claim 15, wherein service data units are stored in a memory part using references.

17. System according to claim 15, provided with a SDU manager.

18. Method for communication between a network device driver program and an application program through intermediate structure software, comprising the steps of:

a. supplying of first data units form the network device driver program to a first program object or protocol object being part of the intermediate structure software;

b. performing of first functions of the first program object on said first data units;

c. supply of resulting second data units from the first program object to a second program object being part of the intermediate structure software;

d. performing of second functions of the second program object on the second data units;

e. supplying of resulting application data units from the second program object to said application program;

wherein supplying data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and

wherein for at least one application data unit, the referenced memory location storing data of the application data unit is the same memory location as the referenced memory location storing at least some of the data of the corresponding first data unit and as the referenced memory location for storing at least some of the data of the corresponding second data unit.

19. Method according to claim 4, wherein within a queue-object two or more priorities for passing of data units are provided.

20. Method according to claim 10, wherein at least two data units referenced by a service data unit are stored in non-contiguous portions of memory.

21. Method according to claim 12, wherein the specialized execution environment forms a plurality of network protocol layers and the first program object and the second program object are in respective network protocol layers.

22. Method according to claim 1, further comprising creating a service data unit for each application data unit, each service data unit including a size value indicating the size of data of the application data unit and an offset value indicating the memory location storing data of the application data unit,

wherein supplying data units between program objects by passing references includes passing service data units corresponding to the supplied data units.

PATENT
450117-4840

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant	:	Yoeri APTS et al.
Serial No.	:	09/246,271
Filed	:	February 8, 1999
For	:	METHOD AND SYSTEM FOR COMMUNICATION BETWEEN APPLICATION PROGRAMS AND A NETWORK
Examiner	:	HO, Andy
Art Unit	:	2126

745 Fifth Avenue
New York, New York 10151

EXPRESS MAIL

Mailing Label Number: ED 521638049 US

Date of Deposit: June 3, 2005

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Chiaki Kokka

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

TRANSMITTAL OF APPELLANTS' BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents,
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith in triplicate is Appellants' Brief in support of their appeal in the above-identified application.

A check in amount of $500.00 is attached in payment of the required fee ($500.00) set forth in Section 1.17(f).

Please charge any additional fees due or credit any overpayment to Deposit Account 50-0320.

Respectfully submitted,

FROMMER, LAWRENCE & HAUG LLP
Attorneys for Appellant

By: Samuel S. Lee, Reg. No. 42,791 for

William S. Frommer
Registration No. 25,506
(212) 588-0800